SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
Program: B.Tech CSDS(311)/B.Tech-CSBS

Course: Machine Learning

Experiment No.01

PART A

(PART A : TO BE REFFERED BY STUDENTS)

A.1 Aim: Introduction to Machine Learning and Pandas library

Task 1: Create a group of two/three students and identify two/three papers based on Machine Learning Applications. The papers must be starting from 2018 onwards (Google Scholar\IEEE). It is mandatory to have papers on at least three different applications. The summary should be strictly in your own words.

i. Paper Title

ii. ML application

iii. Category of ML application (Supervised or Unsupervised)

iv. Your reasoning for category of ML application

v. Algorithms used

vi. Key concepts/ short summary in your own words

Task 2: Perform Exploratory data analysis on Indian cuisine dataset and write the inferences for each question.

i. Read the indianfood1.csv file into a DataFrame.
ii. Explore size, shape, data types of each column in the dataset.
iii. How many total Indian dishes are there?
iv. Using Describe function, view the basic statistics of all columns. What Inference you can make out form that?
v. Are there any missing values in the dataset? If Yes, replace the missing values with the NaN values.
vi. How many numeric features and categorical features are there in the dataset?
vii. Display the number of unique values in each column.
viii. Add a new column in the dataset to calculate the total time taken to make every dish.
ix. Add a new column in the dataset that will count the number of ingredients from the ingredients column for each dish.

A.2 Prerequisite:

Python Programming, Pandas library

A.3 Outcome:

After successful completion of this experiment students will be able to:

i. Differentiate applications of supervised and unsupervised learning
ii. Read different types of data files(csv, excel, text file etc.)
iii. Obtain metadata of given dataset

A.4 Theory:

Machine Learning:

- Definition by Tom Mitchell (1998):
    - Machine Learning is the study of algorithms that improve their performance P at some task T with experience E.
    - A well-defined learning task is given by <P,T,E>
    - For example: Task (T): identifying correct shape
    - Experience (E): images of various shapes as an input to the algorithm
    - Performance (P): number of correctly identified shapes out of all the shapes

Supervised Machine Learning:

Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Basically supervised learning is when we teach or train the machine using data that is well labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labelled data.

Unsupervised Machine Learning:

Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore the machine is restricted to find the hidden structure in unlabeled data by itself.

Exploratory Data Analysis:

Exploratory Data Analysis (EDA) is an open-ended process where we calculate statistics and make figures to find trends, anomalies, patterns, or relationships within the data. The goal of EDA is to learn what our data can tell us. It generally starts out with a high level overview, then narrows in to specific areas as we find intriguing areas of the data. The findings may be interesting in their own right, or they can be used to inform our modeling choices, such as by helping us decide which features to use.

Pandas Library:

key features and functionalities of the Pandas library:

1. DataFrame: The core data structure in Pandas is the DataFrame, which is a two-dimensional, tabular data structure resembling a spreadsheet or SQL table. It organizes data into rows and columns, and you can think of it as a dictionary of Series objects.
2. Series: A Series is a one-dimensional labeled array in Pandas, and it is the building block of a DataFrame. It is similar to a NumPy array but has additional functionality and a labeled index, allowing for more flexible and intuitive data manipulation.

3. Reading and Writing Data: Pandas supports reading and writing data from various file formats, including CSV, Excel, SQL databases, and more. The read_csv(), read_excel(), read_sql(), and related functions make it easy to import data into a DataFrame.
4. Data Cleaning: Pandas provides various methods to handle missing data, duplicate rows, and data manipulation tasks. You can use functions like dropna(), fillna(), drop_duplicates(), and more for data cleaning.
5. Data Selection and Slicing: Pandas allows you to access, slice, and filter data efficiently using labels, row indices, and conditional selections. You can use indexing, boolean masks, and various selection methods like loc[], iloc[], and boolean indexing.
6. Grouping and Aggregation: Pandas offers powerful tools for grouping data based on specific columns and performing aggregate operations like sum, mean, count, etc., on grouped data using groupby() and agg() functions.
7. Merging and Joining Data: You can merge multiple DataFrames based on common columns using functions like merge() and concat(), enabling you to combine data from different sources.
8. Time Series Functionality: Pandas has robust support for time series data, providing features like date/time parsing, resampling, time zone handling, and more.
9. Data Visualization: While Pandas itself does not handle data visualization, it integrates well with popular data visualization libraries like Matplotlib and Seaborn, making it easy to create insightful plots and charts from DataFrame data.

PART B

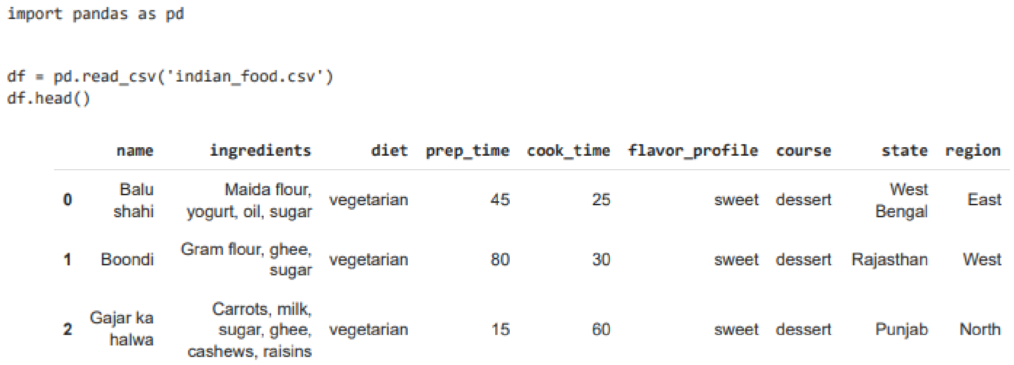
(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical.)

Roll No. A101	Name: Shreyas A. Bailkar
Class : B tech CSBS	Batch : -
Date of Experiment: 22/07/2023	Date of Submission : 23/07/2023
Grade :	

Tasks:

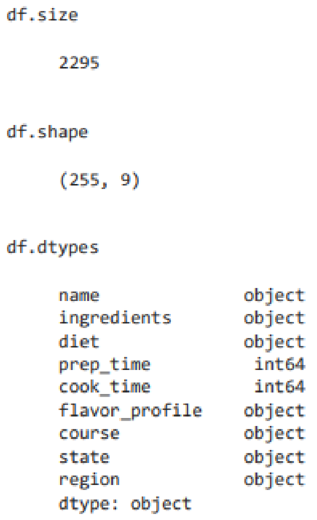
i. Read the indianfood1.csv file into a DataFrame.



```
import pandas as pd

df = pd.read_csv('indian_food.csv')
df.head()
```

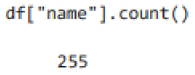
	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	region
0	Balu shahi	Maida flour, yogurt, oil, sugar	vegetarian	45	25	sweet	dessert	West Bengal	East
1	Boondi	Gram flour, ghee, sugar	vegetarian	80	30	sweet	dessert	Rajasthan	West
2	Gajar ka halwa	Carrots, milk, sugar, ghee, cashews, raisins	vegetarian	15	60	sweet	dessert	Punjab	North

ii. Explore size, shape, data types of each column in the dataset.



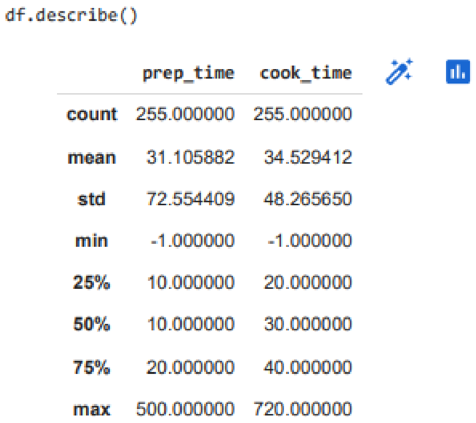
```
df.size

    2295


df.shape

    (255, 9)


df.dtypes

    name              object
    ingredients       object
    diet              object
    prep_time          int64
    cook_time          int64
    flavor_profile    object
    course            object
    state             object
    region            object
    dtype: object
```

iii. How many total Indian dishes are there?



```
df["name"].count()

    255
```

iv. Using Describe function, view the basic statistics of all columns. What Inference you can make out form that?

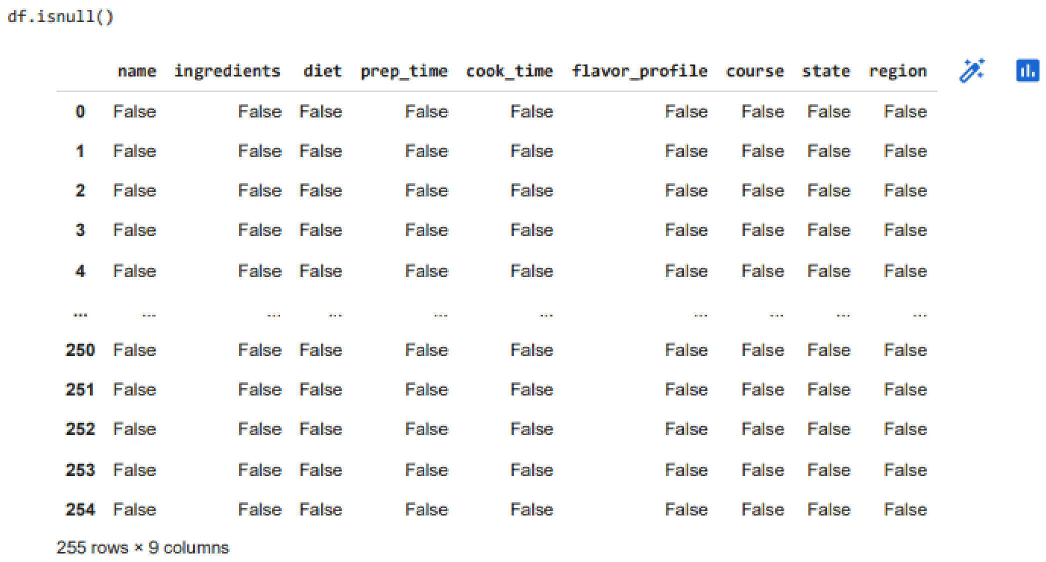


```
df.describe()
```

	prep_time	cook_time
count	255.000000	255.000000
mean	31.105882	34.529412
std	72.554409	48.265650
min	-1.000000	-1.000000
25%	10.000000	20.000000
50%	10.000000	30.000000
75%	20.000000	40.000000
max	500.000000	720.000000

Except ‘prep_time’ and ‘cook_time’, rest all features were object that’s why there stats can’t be defined.

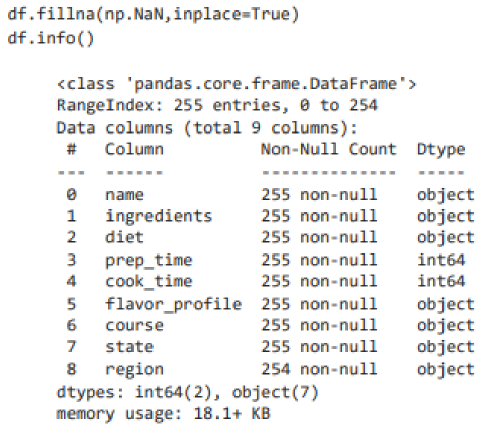
v. Are there any missing values in the dataset? If Yes, replace the missing values with the NaN values.



```
df.isnull()
```

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	region
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
250	False	False	False	False	False	False	False	False	False
251	False	False	False	False	False	False	False	False	False
252	False	False	False	False	False	False	False	False	False
253	False	False	False	False	False	False	False	False	False
254	False	False	False	False	False	False	False	False	False

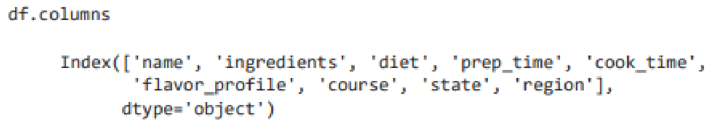
255 rows × 9 columns



```
df.fillna(np.NaN,inplace=True)
df.info()
```

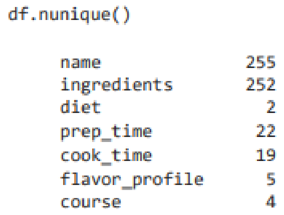
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255 entries, 0 to 254
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   name            255 non-null    object
 1   ingredients     255 non-null    object
 2   diet            255 non-null    object
 3   prep_time       255 non-null    int64 
 4   cook_time       255 non-null    int64 
 5   flavor_profile  255 non-null    object
 6   course          255 non-null    object
 7   state           255 non-null    object
 8   region          254 non-null    object
dtypes: int64(2), object(7)
memory usage: 18.1+ KB
```

vi. How many numeric features and categorical features are there in the dataset?



```
df.columns
```

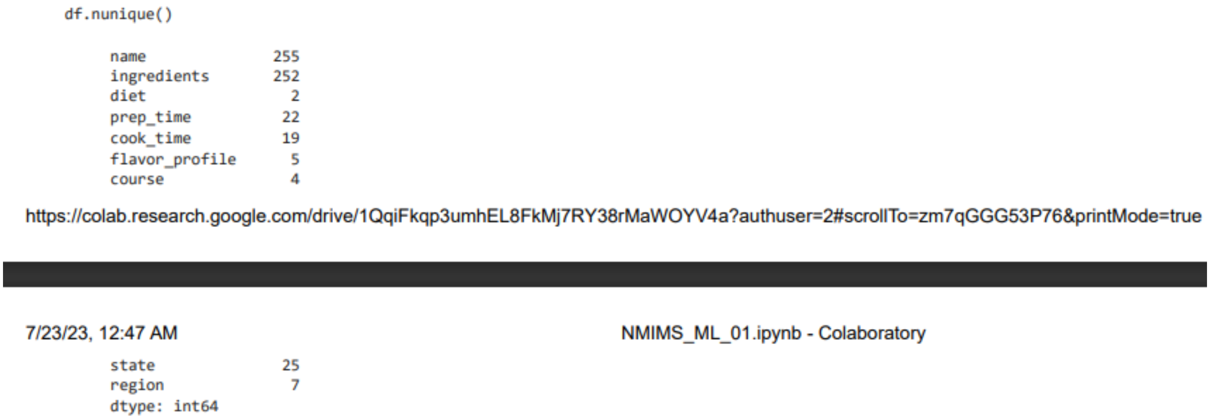
```
Index(['name', 'ingredients', 'diet', 'prep_time', 'cook_time',
       'flavor_profile', 'course', 'state', 'region'],
      dtype='object')
```



```
df.nunique()
```

```
name              255
ingredients       252
diet                2
prep_time          22
cook_time          19
flavor_profile      5
course              4
```

vii. Display the number of unique values in each column.



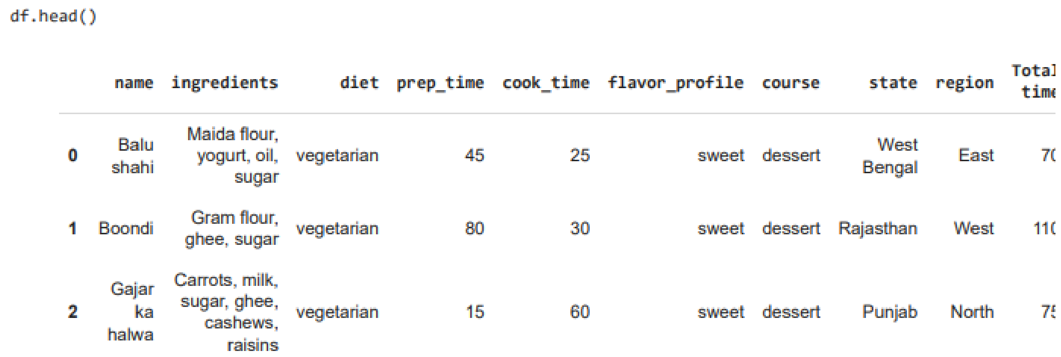
```
df.nunique()

    name             255
    ingredients      252
    diet               2
    prep_time         22
    cook_time         19
    flavor_profile     5
    course             4
```

https://colab.research.google.com/drive/1QqiFkqp3umhEL8FkMj7RY38rMaWOYV4a?authuser=2#scrollTo=zm7qGGG53P76&printMode=true

7/23/23, 12:47 AM NMIMS_ML_01.ipynb - Colaboratory

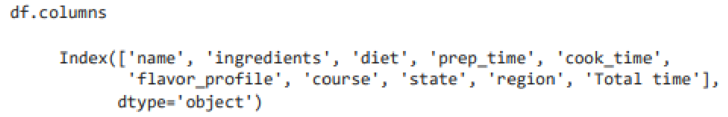
```
    state             25
    region             7
    dtype: int64
```

viii. Add a new column in the dataset to calculate the total time taken to make every dish.



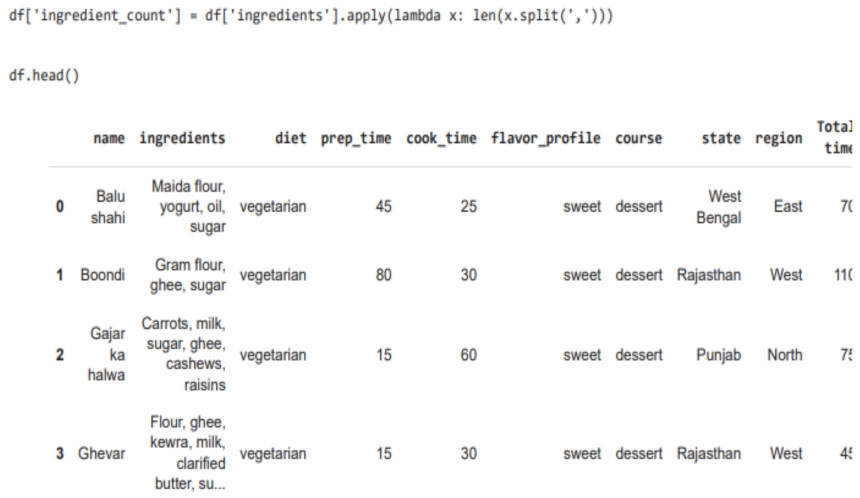
```
df.head()
```

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	region	Total time
0	Balu shahi	Maida flour, yogurt, oil, sugar	vegetarian	45	25	sweet	dessert	West Bengal	East	70
1	Boondi	Gram flour, ghee, sugar	vegetarian	80	30	sweet	dessert	Rajasthan	West	110
2	Gajar ka halwa	Carrots, milk, sugar, ghee, cashews, raisins	vegetarian	15	60	sweet	dessert	Punjab	North	75



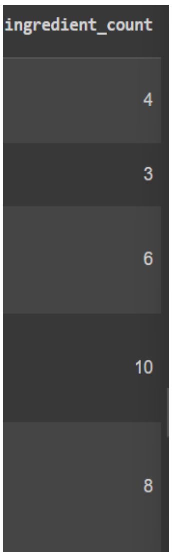
```
df.columns

     Index(['name', 'ingredients', 'diet', 'prep_time', 'cook_time',
            'flavor_profile', 'course', 'state', 'region', 'Total time'],
           dtype='object')
```

ix. Add a new column in the dataset that will count the number of ingredients from the ingredients column for each dish.



```
df['ingredient_count'] = df['ingredients'].apply(lambda x: len(x.split(',')))

df.head()
```

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	region	Total time
0	Balu shahi	Maida flour, yogurt, oil, sugar	vegetarian	45	25	sweet	dessert	West Bengal	East	70
1	Boondi	Gram flour, ghee, sugar	vegetarian	80	30	sweet	dessert	Rajasthan	West	110
2	Gajar ka halwa	Carrots, milk, sugar, ghee, cashews, raisins	vegetarian	15	60	sweet	dessert	Punjab	North	75
3	Ghevar	Flour, ghee, kewra, milk, clarified butter, su...	vegetarian	15	30	sweet	dessert	Rajasthan	West	45



ingredient_count
4
3
6
10
8

Conclusion:

With this hands-on experimentation, we got an understanding of pandas, numpy and matplotlib libraries in python.

Note: Pdf of Google colab is attached